

SLAV: Test stack abstraction layers

Paweł Wieczorek

October 31, 2019

Samsung R&D Institute Poland

Motivation

Testing laboratory layers

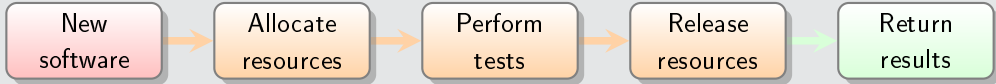
Case study

Summary

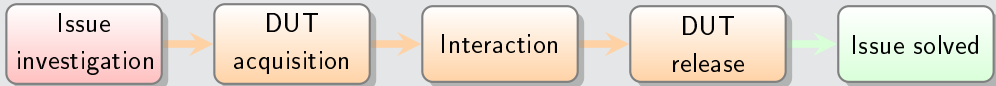
Motivation

Use cases

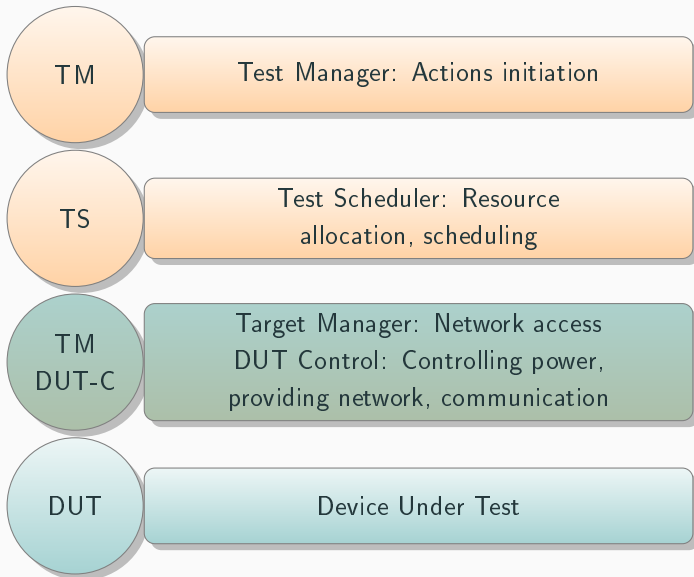
Automated testing



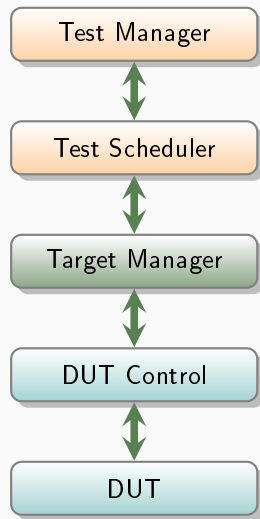
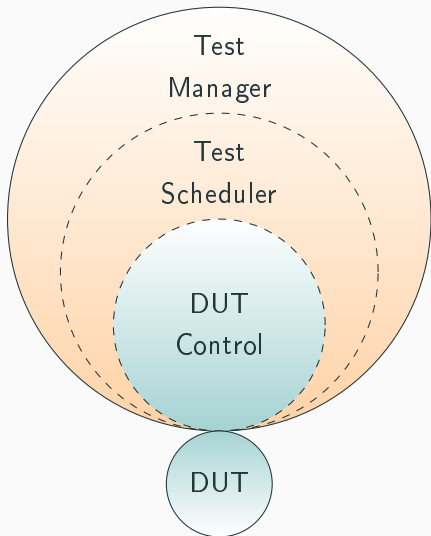
Direct access (*hacking*)



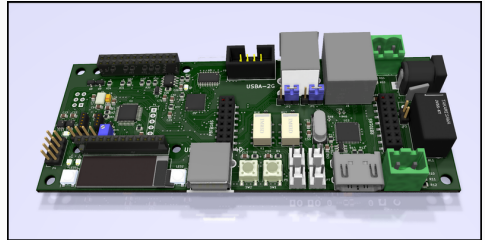
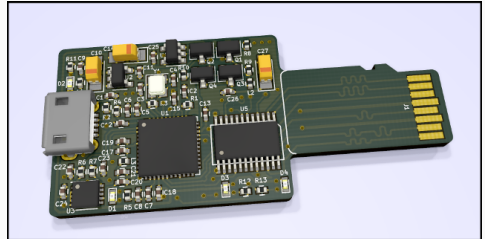
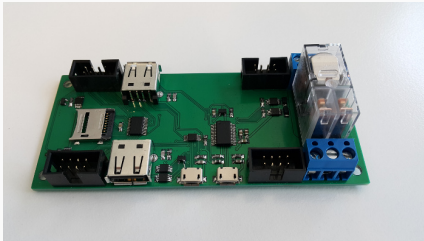
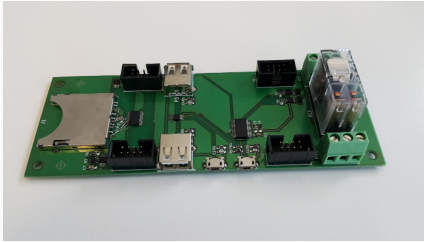
Abstractions (https://elinux.org/Test_Standards)



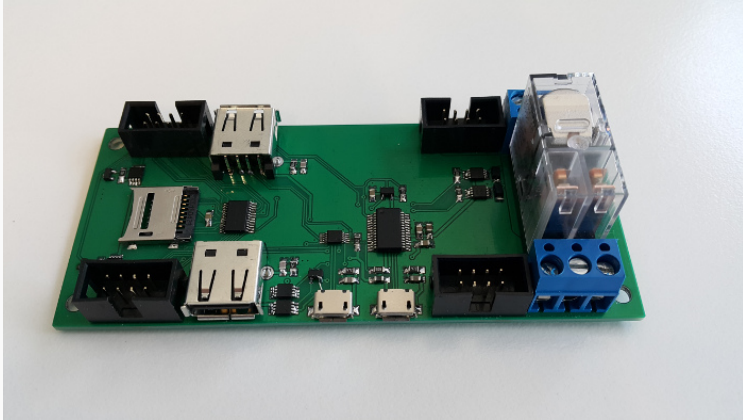
Laboratory structure



Devices (for DUT Control and Target Management)

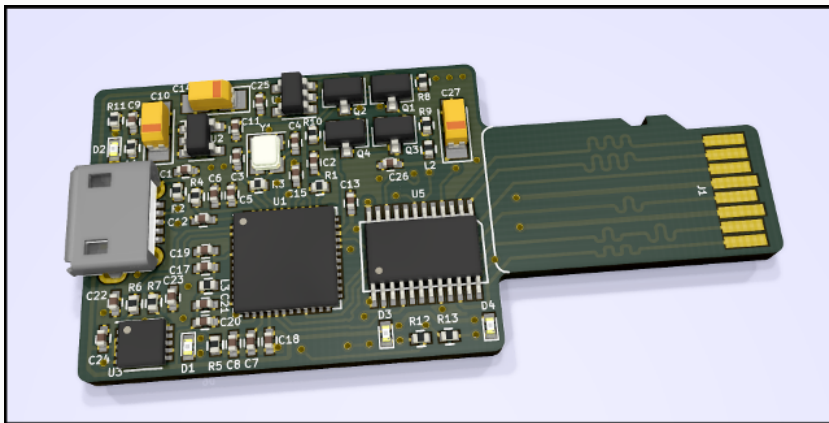


SD-MUX (DUT Control)



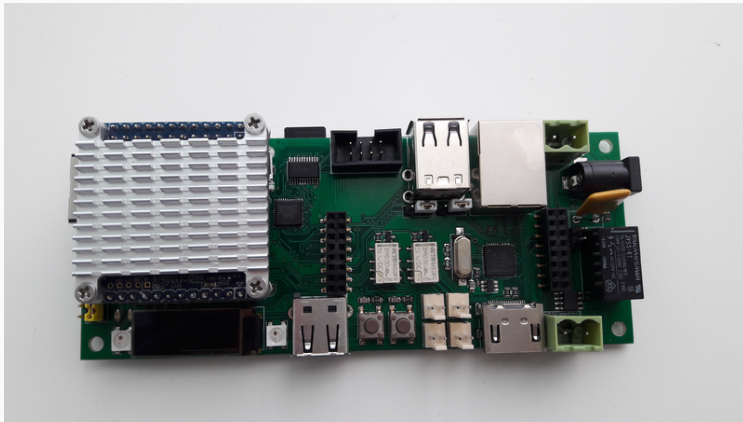
https://wiki.tizen.org/SD_MUX

SDWire (DUT Control)



<https://wiki.tizen.org/SDWire>

MuxPi (DUT Control and Target Management)



<https://wiki.tizen.org/MuxPi>

Testing laboratory layers

Challenges

Knowledge

Which actions are necessary? Where can it be performed? How to do it?

Responsibilities

Who performs given action?

Sharing

Who can use DUT? How can DUT be used?

Implementation

Test Manager (minimal)

- initiate actions
- list (or cancel) currently performed actions

Test scheduler (generic)

- list available resources, request specific ones
- acquire assigned resources (then prolong, finally release)

DUT Control (*tricky*)

- boot (and login)
- execute commands
- copy files

Test Manager (<https://github.com/SamsungSLAV/weles>)

API path	Feature
/jobs	Create a new <i>job</i>
/jobs/list	List all <i>jobs</i>
/jobs/:id	Get information on <i>job</i>
/jobs/:id/cancel	Cancel given <i>job</i>

- Receives simple task scenarios
- Requests and acquires access to DUT (via Target Manager), *flashes SD card*, executes Target Manager API commands

Test Scheduler (<https://github.com/SamsungSLAV/boruta>)

API path	Feature
/reqs	Create a new <i>request</i>
/reqs/list	List all <i>requests</i>
/reqs/:id	Get information on <i>request</i>
/reqs/:id/acquire	Acquire access to Target Manager satisfying <i>request</i>
/reqs/:id/prolong	Prolong access to Target Manager
/reqs/:id/close	Release Target Manager
/workers/list	List all Target Managers
/workers/:id	Get information on Target Manager (esp. <i>capabilities</i>)
/workers/:id/setstate	(Admin) “IDLE”, “MAINTENANCE” (+ “FAILED”)
/workers/:id/setgroups	(Admin) Set <i>groups</i> for Target Manager
/workers/:id/deregister	(Admin) Remove Target Manager from the pool

Target Manager (Boruta + <https://github.com/SamsungSLAV/muxpi>) 1/2

```
listen_address = ":7175"
```

```
boruta_address = ""
```

```
ssh_address = ":22"
```

```
[caps]
```

```
    UUID = "901a513b-0166-4786-bb35-68ea3c194f57"
```

```
    device_type = "qemu"
```

```
[user]
```

```
    name = "boruta-user"
```

- Reports target state, providing access to DUT (over SSH)
- Provides actual connection to DUT (network, serial), controls power and SD mux

Target Manager (Boruta + <https://github.com/SamsungSLAV/muxpi>) 2/2

<i>API path</i> (on Target Manager)	Feature
<code>/usr/local/bin/dut_boot.sh</code>	Boot DUT
<code>/usr/local/bin/dut_login.sh</code>	Login to DUT shell (in case of serial)
<code>/usr/local/bin/dut_exec.sh</code>	Execute command on DUT
<code>/usr/local/bin/dut_copyto.sh</code>	Copy files to DUT
<code>/usr/local/bin/dut_copyfrom.sh</code>	Copy files from DUT
<code>/usr/local/bin/fota</code>	Flash over-the-air (SD mux-specific)
<code>/usr/local/bin/stm</code>	DUT Control (MuxPi-specific)

Case study

Strengths

- Requires only preparing test plan
- Test plans can be reused among various projects

Weaknesses

- Keeping compliance
- Catching up with others (e.g. LAVA, SQUAD)

Strengths

- Users treated equally
- Resource type-agnostic

Weaknesses

- Requires additional agent
- Capabilities declared up front

Strengths

- Only some knowledge required
- Unification possibility

Weaknesses

- Hard initial setup
- Often unique for a given testing laboratory

Summary

- Unable to demo without specific hardware
- Risky large scale deployments
- Responsibilities division allows *easier* onboarding
- Multiple repositories made integration unnecessarily difficult

- User-centric approach resulted in smaller building blocks
- Smaller blocks could be easier swapped or used independently
- Improvement needs more *reuse* instead of *rewrite*

<https://github.com/SamsungSLAV/>*

Thank you!

Paweł Wieczorek
p.wieczorek2@samsung.com